UNITED STATES PATENT AND TRADEMARK OFFICE

APPLICATION NO.	FILING DATE	FIRST NAMED INVENTOR	ATTORNEY DOCKET NO.	CONFIRMATION NO.
10/729,767	12/08/2003	Richard P. Himmer	I20 04983US	3406

128   7590   03/13/2008
HONEYWELL INTERNATIONAL INC.
101 COLUMBIA ROAD
P O BOX 2245
MORRISTOWN, NJ 07962-2245

EXAMINER
WANG, BEN C

ART UNIT	PAPER NUMBER
2192	

MAIL DATE	DELIVERY MODE
03/13/2008	PAPER

Please find below and/or attached an Office communication concerning this application or proceeding.

The time period for reply, if any, is set in the attached communication.

-- The MAILING DATE of this communication appears on the cover sheet with the correspondence address --

Period for Reply

A SHORTENED STATUTORY PERIOD FOR REPLY IS SET TO EXPIRE *3* MONTH(S) OR THIRTY (30) DAYS, WHICHEVER IS LONGER, FROM THE MAILING DATE OF THIS COMMUNICATION.
- Extensions of time may be available under the provisions of 37 CFR 1.136(a). In no event, however, may a reply be timely filed after SIX (6) MONTHS from the mailing date of this communication.
- If NO period for reply is specified above, the maximum statutory period will apply and will expire SIX (6) MONTHS from the mailing date of this communication.
- Failure to reply within the set or extended period for reply will, by statute, cause the application to become ABANDONED (35 U.S.C. § 133). Any reply received by the Office later than three months after the mailing date of this communication, even if timely filed, may reduce any earned patent term adjustment. See 37 CFR 1.704(b).

Status

1)☒ Responsive to communication(s) filed on *04 December 2007*.

2a)☐ This action is **FINAL.**   2b)☒ This action is non-final.

3)☐ Since this application is in condition for allowance except for formal matters, prosecution as to the merits is closed in accordance with the practice under *Ex parte Quayle*, 1935 C.D. 11, 453 O.G. 213.

Disposition of Claims

4)☒ Claim(s) *1-8, 16 and 17* is/are pending in the application.

    4a) Of the above claim(s) _____ is/are withdrawn from consideration.

5)☐ Claim(s) _____ is/are allowed.

6)☒ Claim(s) *1-8 and 16-17* is/are rejected.

7)☐ Claim(s) _____ is/are objected to.

8)☐ Claim(s) _____ are subject to restriction and/or election requirement.

Application Papers

9)☐ The specification is objected to by the Examiner.

10)☐ The drawing(s) filed on _____ is/are: a)☐ accepted or b)☐ objected to by the Examiner.

    Applicant may not request that any objection to the drawing(s) be held in abeyance. See 37 CFR 1.85(a).

    Replacement drawing sheet(s) including the correction is required if the drawing(s) is objected to. See 37 CFR 1.121(d).

11)☐ The oath or declaration is objected to by the Examiner. Note the attached Office Action or form PTO-152.

Priority under 35 U.S.C. § 119

12)☐ Acknowledgment is made of a claim for foreign priority under 35 U.S.C. § 119(a)-(d) or (f).

    a)☐ All   b)☐ Some * c)☐ None of:

        1.☐ Certified copies of the priority documents have been received.

        2.☐ Certified copies of the priority documents have been received in Application No. _____.

        3.☐ Copies of the certified copies of the priority documents have been received in this National Stage application from the International Bureau (PCT Rule 17.2(a)).

    * See the attached detailed Office action for a list of the certified copies not received.

Attachment(s)

1) ☒ Notice of References Cited (PTO-892)
2) ☐ Notice of Draftsperson's Patent Drawing Review (PTO-948)
3) ☐ Information Disclosure Statement(s) (PTO/SB/08)
    Paper No(s)/Mail Date _____.

4) ☐ Interview Summary (PTO-413)
    Paper No(s)/Mail Date. _____.
5) ☐ Notice of Informal Patent Application
6) ☐ Other: _____.

DETAILED ACTION

1.   Applicant's amendment dated December 4, 2007, responding to the Office action

mailed September 4, 2007 provided in the rejection of claims 1-10 and 16-17, wherein

claims 1, 5, 8 and 17 are amended, claims 9-10 are canceled.

Claims 1-8 and 16-17 remain pending in the application and which have been

fully considered by the examiner.

Applicant's arguments with respect to claims currently amended have been fully

considered but are moot in view of the new grounds of rejection – see *Hsiao et al.* - art made

of record, as applied hereto.

Claim Objections

Claims 5, 8, and 17 are objected to because the following informalities

- Status of currently amended claims 5, 8, and 17 were inadvertently listed as

  "Original" should be corrected as --Currently amended-- instead.

Appropriate correction is required.

Claim Rejections – 35 USC § 103(a)

The following is a quotation of 35 U.S.C. 103(a) which forms the basis for all

obviousness rejections set forth in this office action:

> (a) A patent may not be obtained though the invention is not identically disclosed or described as set
> forth in section 102 of this title, if the differences between the subject matter sought to be patented
> and the prior art are such that the subject matter as a whole would have been obvious at the time the

invention was made to a person having ordinary skill in the art to which said subject matter pertains. Patentability shall not be negatived by the manner in which the invention was made.

2.   Claims 1-8 and 17 are rejected under 35 U.S.C. 103(a) as being unpatentable over Hammack et al. (Pat. No. US 6,449,624 B1) in view of Hsiao et al. (*Integrated XML Document Management, January 1, 2003, Springer-Verlag Berlin Heidelberg*) (hereinafter 'Hsiao' - art made of record)

3.   **As to claim 1** (Currently Amended), Hammack discloses a source control system for a process control system (e.g., Fig. 1 – a process control system; Col. 3, Lines 49-52 – a process control system includes a process controller connected to one or more host workstations or computers via a communication network such as an Ethernet connection or the like), comprising:

- a processor in a process control system (e.g., Col. 2, Lines 36-39 – the inventive system further includes a configuration routine and a version control routine, both of which are stored in the computer-readable medium and configured to be executed by the processor);

- a database accessible by said processor to store information associated with an object under source control to be checked-out (e.g., Col. 12, Lines 66 through Col. 13, Lines 7 – the VCAT [Version Control and Audit Trail system] system may further include functionality that permits the user to request a status update for all of the items in the configuration database to ensure that those items that are checked out are indicated as such via checkmark or the like); and

- a check-out function operable on said processor to check-out said object (e.g.,

    Fig. 6 – element of 120 – "Check Out" , "CheckOut Recursive"); to use said

    information to determine whether any dependent objects exist, and to

    automatically check-out said existing dependent objects (e.g., Col. 11, Lines 24-

    29 – the VCAT system preferably determines during each check-out operation

    which other versionable items need to be checked out in order to modify the

    configuration of an item; the modification of these other versionable items may be

    referred to as "consequential changes.", 45-48 – because the configuration of the

    process is set forth in a hierarchal manner, the VCAT system must allow for

    checking out items having subordinate items that are also versionable).

Hammack discloses the data stored in an XML document is accessed in accordance

with an object model that provides for parsing the document to create a data tree

structure having a plurality of nodes associated with the version control data (e.g., Col.

22, Lines 52-58), but does not explicitly disclose wherein said object is a user defined

template, and wherein said existing dependent objects are children user defined

templates of said object or instances of said object or of said children user defined

templates.

However, in an analogous art of *Integrated XML Document Management*, Hsiao

discloses wherein said object is a user defined template, and wherein said existing

dependent objects are children user defined templates of said object or instances of

said object or of said children user defined templates (e.g., Sec. 3.2 - Metadata Server

Data Model, 1st Par - ... An item is the basic unit of resource .... An item is a typed

object whose type is defined <u>by an Item Type</u> … <u>an Item Type</u> comprises <u>Component

Types</u> <u>arranged in a hierarchy</u>. This hierarchy forms <u>a tree structure</u> and a unique root

Component Type. An Item is <u>an instance of an Item Type</u>. It comprises <u>one instance of

the root Component Type</u> and <u>zero or more instances of descendent Component

Types</u>… Within the Item, these component instances have <u>ancestor-descendent

relationships</u> as dictated by the Item Type definition … Items can be associated with

one another via links. A link describes a relationship between two Items or a relationship

of an Item to itself; Fig. 2(a) – Example of Library Server data model; Sec. 3.2 -

Metadata Server Data Model, 2nd Par., Lines 6-7 – The *ComponentId* and

ParentCompoentId columns capture <u>parent-child relationships between components</u>).

Therefore, it would have been obvious to one of ordinary skill in the art, at the

time the invention was made to combine the teachings of Hsiao into the Hammack's

system to further provide wherein said object is a user defined template, and wherein

said existing dependent objects are children user defined templates of said object or

instances of said object or of said children user defined templates in the Hammack

system.

The motivation is that it would further enhance the Hammack's system by taking,

advancing and/or incorporating Hsiao's system which offers significant advantages to

have the metadata server separated from the data server such that a metadata server

can be associated with multiple data servers and each of the data servers can be

placed close to the set of users that most frequently access a server; this allows

centralized searching with distributed and efficient data delivery to users as once

suggested by Hsiao (e.g., Sec. 1 Introduction, 2nd Par., Lines 4-10)

4.   **As to claim 2** (incorporating the rejection in claim 1) (Original), Hammack

discloses the system further comprising: a propagation function operable on said

processor to propagate changes made to said object to said existing dependent objects,

when said object is saved (e.g., Col. 29, Lines 11-16 – the configuration management

system wherein the configuration routine is adapted to be executed by the processor to

make changes to a first process control element and to propagate changes to other

process control elements that are affected by the changes made to the first process

control element).

5.   **As to claim 3** (incorporating the rejection in claim 1) (Original), Hammack

discloses the system wherein said stored information includes a reference to a parent

object (e.g., TABLE 4 – Recover/Purge Dialog Window, 1st entry – items – list all

subordinate items deleted based upon a selected parent; Col. 23, Lines 36-41, 49-56).

6.   **As to claim 4** (incorporating the rejection in claim 1) (Original), Hammack

discloses the system wherein said stored information is at least one selected from the

group consisting of: a name, a version number, a type and a status (e.g., Col. 20, Lines

15-20 – Furthermore, it is preferred that key words and labels be utilized to identify

attributes such as object type and properties; examples of property labels are "NAME"

and "DESCRIPTION"; Col. 13, Lines 58-66 – data representative of each prior

configuration of an item is stored in the version control database together with data

reflective of a version assigned thereto; the version is preferably identified by number,

but may be indicated in any other manner).

7.   **As to claim 5** (Currently Amended), Hammack discloses a method of automatic

check-out for a source control system in a process control system, comprising:

- storing information associated with an object (e.g., Fig. 3; Col. 6, Lines 25-43 –

  with reference now to Fig. 3, the data stored in the configuration database may

  be presented to a user via a configuration database administrative interface such

  as Delta V® Explorer, which will hereinafter be referred to as "the Explorer

  system"; the Explorer system sets forth a configuration hierarchy in a windows-

  type environment having a suite of configuration tools for modifying the elements

  of the hierarchy);

- receiving a request from a user to check-out said object (e.g., Fig. 6 – element of

  120 – "Check Out" , "CheckOut Recursive");

- determining whether any dependent objects of said object exist based on said

  information; automatically checking-out said existing dependent objects when

  said object is checked-out; and providing a status to said user (e.g., Col. 11,

  Lines 24-29 – the VCAT system preferably determines during each check-out

  operation which other versionable items need to be checked out in order to

  modify the configuration of an item; the modification of these other versionable

items may be referred to as "consequential changes.", 45-48 – because the configuration of the process is set forth in a hierarchal manner, the VCAT system must allow for checking out items having subordinate items that are also versionable).

Hammack discloses the data stored in an XML document is accessed in accordance with an object model that provides for parsing the document to create a data tree structure having a plurality of nodes associated with the version control data (e.g., Col. 22, Lines 52-58), but does not explicitly disclose wherein said object is a user defined template, and wherein said existing dependent objects are children user defined templates of said object or instances of said object or of said children user defined templates.

However, in an analogous art of *Integrated XML Document Management*, Hsiao discloses wherein said object is a user defined template, and wherein said existing dependent objects are children user defined templates of said object or instances of said object or of said children user defined templates (e.g., Sec. 3.2 - Metadata Server Data Model, 1st Par - ... An item is the basic unit of resource .... An item is a typed object whose type is defined by an Item Type ... an Item Type comprises Component Types arranged in a hierarchy. This hierarchy forms a tree structure and a unique root Component Type. An Item is an instance of an Item Type. It comprises one instance of the root Component Type and zero or more instances of descendent Component Types... Within the Item, these component instances have ancestor-descendent relationships as dictated by the Item Type definition ... Items can be associated with

one another via links. A link describes a relationship between two Items or a relationship

of an Item to itself; Fig. 2(a) – Example of Library Server data model; Sec. 3.2 -

Metadata Server Data Model, 2nd Par., Lines 6-7 – The *ComponentId* and

ParentCompoentId columns capture <u>parent-child relationships between components</u>).

Therefore, it would have been obvious to one of ordinary skill in the art, at the

time the invention was made to combine the teachings of Hsiao into the Hammack's

system to further provide wherein said object is a user defined template, and wherein

said existing dependent objects are children user defined templates of said object or

instances of said object or of said children user defined templates in the Hammack

system.

The motivation is that it would further enhance the Hammack's system by taking,

advancing and/or incorporating Hsiao's system which offers significant advantages to

have the metadata server separated from the data server such that a metadata server

can be associated with multiple data servers and each of the data servers can be

placed close to the set of users that most frequently access a server; this allows

centralized searching with distributed and efficient data delivery to users as once

suggested by Hsiao (e.g., Sec. 1 Introduction, 2nd Par., Lines 4-10)

8.   **As to claim 6** (incorporating the rejection in claim 5) (Original), Hammack

discloses the method further comprising: sorting said existing dependent objects so that

parents precede children (e.g., Col. 11, Lines 48-51 – in one embodiment, if a recursive

check-out or check-in is selected by the user, the VCAT system generates a dialog

window that provides the user with a list of versionable, subordinate items that may be checked out (or checked-in)).

9.   **As to claim 7** (incorporating the rejection in claim 5) (Original), Hammack discloses the method wherein one of said existing dependent objects is a derivation child of said object (e.g., Col. 11, Lines 45-55 – because the configuration of the process is set forth in a hierarchal manner, the VCAT system must allow for checking out items having subordinate items that are also versionable).

10.   **As to claim 8** (incorporating the rejection in claim 7) (Currently Amended), Hammack discloses the method further comprising: automatically checking-out a derivation child only if said derivation child is checked-in (e.g., Col. 11, Lines 45-55 – because the configuration of the process is set forth in a hierarchal manner, the VCAT system must allow for checking out items having subordinate items that are also versionable).

11.   **As to claim 17** (Currently Amended), Hammack discloses a computer readable medium having executable instructions stored thereon to perform a method of automatic check-out for a source control system in a process control system, said method comprising:

• storing information associated with an object (e.g., Fig. 3; Col. 6, Lines 25-43 – with reference now to Fig. 3, the data stored in the configuration database may

be presented to a user via a configuration database administrative interface such

as Delta V® Explorer, which will hereinafter be referred to as "the Explorer

system"; the Explorer system sets forth a configuration hierarchy in a windows-

type environment having a suite of configuration tools for modifying the elements

of the hierarchy);

- receiving a request from a user to check-out said object (e.g., Fig. 6 – element of

  120 – "Check Out" , "CheckOut Recursive");

- determining whether any dependent objects of said object exist based on said

  information; automatically checking-out said existing dependent objects when

  said object is checked-out; and providing a status to said user (e.g., Col. 11,

  Lines 24-29 – the VCAT system preferably determines during each check-out

  operation which other versionable items need to be checked out in order to

  modify the configuration of an item; the modification of these other versionable

  items may be referred to as "consequential changes.", 45-48 – because the

  configuration of the process is set forth in a hierarchal manner, the VCAT system

  must allow for checking out items having subordinate items that are also

  versionable).

Hammack discloses the data stored in an XML document is accessed in accordance

with an object model that provides for parsing the document to create a data tree

structure having a plurality of nodes associated with the version control data (e.g., Col.

22, Lines 52-58), but does not explicitly disclose wherein said object is a user defined

template, and wherein said existing dependent objects are children user defined

templates of said object or instances of said object or of said children user defined

templates.

However, in an analogous art of *Integrated XML Document Management*, Hsiao

discloses wherein said object is a user defined template, and wherein said existing

dependent objects are children user defined templates of said object or instances of

said object or of said children user defined templates (e.g., Sec. 3.2 - Metadata Server

Data Model, 1st Par - ... <u>An item</u> is the basic unit of resource .... <u>An item</u> is <u>a typed</u>

<u>object</u> whose type is defined <u>by an Item Type</u> ... <u>an Item Type</u> comprises <u>Component</u>

<u>Types</u> <u>arranged in a hierarchy</u>. This hierarchy forms <u>a tree structure</u> and a unique root

Component Type. An Item is <u>an instance of an Item Type</u>. It comprises <u>one instance of</u>

<u>the root Component Type</u> and <u>zero or more instances of descendent Component</u>

<u>Types</u>... Within the Item, these component instances have <u>ancestor-descendent</u>

<u>relationships</u> as dictated by the Item Type definition ... Items can be associated with

one another via links. A link describes a relationship between two Items or a relationship

of an Item to itself; Fig. 2(a) – Example of Library Server data model; Sec. 3.2 -

Metadata Server Data Model, 2nd Par., Lines 6-7 – The *ComponentId* and

ParentCompoentId columns capture <u>parent-child relationships between components</u>).

Therefore, it would have been obvious to one of ordinary skill in the art, at the

time the invention was made to combine the teachings of Hsiao into the Hammack's

system to further provide wherein said object is a user defined template, and wherein

said existing dependent objects are children user defined templates of said object or

instances of said object or of said children user defined templates in the Hammack

system.

The motivation is that it would further enhance the Hammack's system by taking,

advancing and/or incorporating Hsiao's system which offers significant advantages to

have the metadata server separated from the data server such that a metadata server

can be associated with multiple data servers and each of the data servers can be

placed close to the set of users that most frequently access a server; this allows

centralized searching with distributed and efficient data delivery to users as once

suggested by Hsiao (e.g., Sec. 1 Introduction, 2nd Par., Lines 4-10)

Claim Rejections – 35 USC § 102(e)

The following is quotation of 35 U.S.C. 102(e) which form the basis for all obviousness

rejections set forth in this office action:

> (e) the invention was described in (1) an application for patent, published under section 122(b), by another filed
> in the United States before the invention by the applicant for patent or (2) a patent granted on an application for
> patent by another filed in the United States before the invention by the applicant for patent, except that an
> international application filed under the treaty defined in section 351(a) shall have the effects for purposes of this
> subsection of an application filed in the United States only if the international application designated the United
> States and was published under Article 21(2) of such treaty in the English language.

12.   Claim 16 is rejected under 35 U.S.C. 102(e) as being anticipated by Hsiao

13.   **As to claim 16** (Previously Presented), Hsiao discloses a computer readable

medium having executable instructions stored thereon to perform a method of

determining object relationships when checking-in, said method comprising:

- determining whether an object to be checked-in has a first derivation parent;

  adding a name and a version of said first derivation parent to a list of object

  relationships, if said object has said first derivation parent (e.g., Sec. 3.2 -

  Metadata Server Data Model, 1st Par - ... An item is the basic unit of resource

  .... An item is a typed object whose type is defined by an Item Type … an

  Item Type comprises Component Types arranged in a hierarchy. This

  hierarchy forms a tree structure and a unique root Component Type. An Item

  is an instance of an Item Type. It comprises one instance of the root

  Component Type and zero or more instances of descendent Component

  Types… Within the Item, these component instances have ancestor-

  descendent relationships as dictated by the Item Type definition … Items can

  be associated with one another via links. A link describes a relationship

  between two Items or a relationship of an Item to itself; Fig. 2(a) – Example of

  Library Server data model; Sec. 3.2 - Metadata Server Data Model, 2nd Par.,

  Lines 6-7 – The *ComponentId* and *ParentCompoentId* columns capture

  parent-child relationships between components);

- determining for each contained object that is contained in said object, whether

  said contained object has a second derivation parent, if said object does not

  have said first derivation parent; adding a name and a version of said second

  derivation parent to said list of object relationships, if said contained object

  has said second derivation parent; and providing said list of object

  relationships (e.g., Sec. 4.3 - Integrated Query, 1st Par., Lines 6-12 – the

nesting of XML elements mimics the hierarchy of components within an Item.

The attributes of components are represented by XML attributes of the

corresponding XML element. A Link relationship is represented by an

<OutboundLink> element nested with the source Item of the Link and also by

an <InboundLink> XML element nested within the target Item of the Link …;

2nd Par. - … Representing link relationships as XML elements nested within

the XML elements representing the source and the target Items ….).

Conclusion

14.   Any inquiry concerning this communication or earlier communications from the examiner should be directed to Ben C. Wang whose telephone number is 571-270-1240. The examiner can normally be reached on Monday - Friday, 8:00 a.m. - 5:00 p.m., EST.

If attempts to reach the examiner by telephone are unsuccessful, the examiner's supervisor, Tuan Q. Dam can be reached on 571-272-3695. The fax phone number for the organization where this application or proceeding is assigned is 571-273-8300.

Information regarding the status of an application may be obtained from the Patent Application Information Retrieval (PAIR) system. Status information for published applications may be obtained from either Private PAIR or Public PAIR. Status information for unpublished applications is available through Private PAIR only. For more information about the PAIR system, see http://pair-direct.uspto.gov. Should you have questions on access to the Private PAIR system, contact the Electronic Business Center (EBC) at 866-217-9197 (toll-free). If you would like assistance from a USPTO Customer Service Representative or access to the automated information system, call 800-786-9199 (IN USA OR CANADA) or 571-272-1000.

/Ben C Wang/
Examiner, Art Unit 2192
February 26, 2008

/Tuan Q. Dam/

Supervisory Patent Examiner, Art Unit 2192